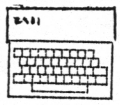
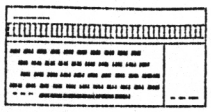



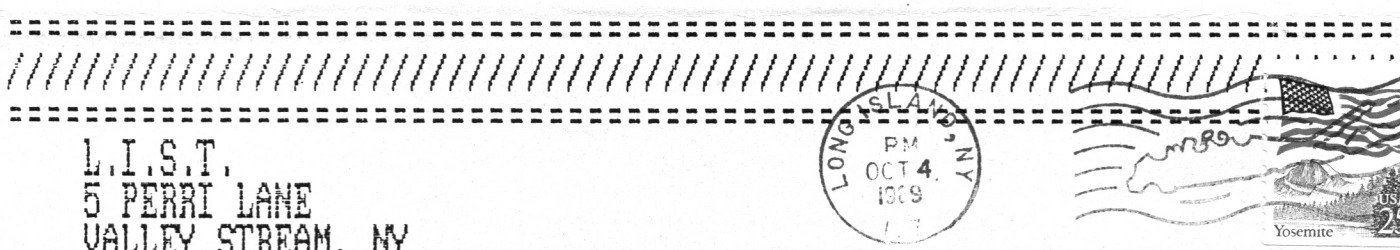
# LIST

LONG ISLAND SINCLAIR TIMEX GROUP
INCORPORATING * NYTSE OF NEW YORK CITY
ISSUE: **September** 1989

DISK DRIVES SUPPORTED:
1 OLIGER
2 LARKEN
3 AERCO

LIST MEMBERSHIP IS $15.00. LIBRARY TAPES ARE AVALIABLE, WRITE THE ADDRESS PRINTED BELOW.


## Software Review Software Review
## Hardware Review Hardware Review

L.I.S.T.
5 PERRI LANE
VALLEY STREAM, NY
11581







TO:

Don Lambert JAN/90
3310 Clover Dr. S
Cedar Rapids, IA
52404

FIRST CLASS MAIL
DATED MEETING NOTICE

UPPER RIGHT CORNER OF YOUR LABEL IS DATE OF LAST ISSUE.

## LIST OFFICERS

| | |
|---|---|
| PRES. | HARVEY RAIT |
| TRES. | ROBERT MOLLOY |
| REC.SEC. | STEVE KAYE |
| EDITOR. | FRED STERN |
| LIBR. | TOM SKAPINSKI |

PLEASE SEND INQUIRIES TO:
LIST
MR. HARVEY RAIT
5 PERI LANE
VALLEY STREAM, N.Y. 11581

PLEASE SEND SUBMISSIONS TO:
LIST
MR. FRED STERN
214 ROBERT ST.
HOLBROOK, N.Y. 11741

## COMING EVENTS:

OCT. 18, 1988 LIST MEETING
OCT 18, 1988 NYTSE MEETING
OCT 22, 1988 COMPUTER SHOW AND FLEAMARKET

## MEETING MINUTES
## SEPT 10, 1988

THE MEETING WAS CALLED TO ORDER AT 2:15 BY HARVEY

THE MAIL RECEIVED DURING THE SUMMER WAS READ. FRED STERN WILL ANSWER ENQUIRIES ABOUT JOINING LIST.

LIST TAPES 1 AND 2 (TS1000 PROGRAM) ARE IN PROCESS OF HAVING NEW MASTERS OUT.

TS1000 TECHNICAL TIDBITS IS READY FOR SALE. THE PRICE OF $4.00 A COPY, POST PAID WAS DETERMINED.

A TAPE LIBRARIAN FOR THE TS2068 AND SPECTRUM TAPES IS NEEDED. BOB MOLLOY WILL REVIEW AND CATALOG THE TAPE LIBRARY ON AN INTERUM BASIS.

AN ALTERNATE MEETING PLACE IS NEEDED, PREFERABLY IN NASSAU COUNTY. ANY MEMBER WHO CAN HELP PLEASE CONTACT HARVEY ASAP.

OCT. 22, 1988, A COMPUTER SHOW AND FLEAMARKET WILL BE HELD AT THE ROYCE CARLIN HOTEL, RTE 110, MELVILLE, N.Y.

BOB GILDER DEMONSTRATED A FULL STROKE (QWERTY) KEYBOARD HE BUILT FOR THE Q.L.

TOM SKAPINSKI GAVE A REVIEW OF (PRINT FACTORY) SOFTWARE FROM BYTE POWER. THIS SOFTWARE, FOR THE TS2068, RECEIVED A HIGH GRADE FROM TOM. THE NEW FRONT COVER LOGO OF THIS EDITION OF LISTING WAS DONE BY TOM USING PRINT FATORY.

A $1.00 ENTRANCE FEE WILL BE CHARGED TO ALL MEMBERS AND GUESTS ATTENDING THE MEETING TO HELP OFFSET THE COST OF REFRESHMENTS.

---

THE QL MICROCASSETTES ARE IN. THEY ARE SOLD AS A PACKAGE OF FOUR (4) IN A WALLET STYLE HOLDER FOR $10.00. SEE HARVEY BEFORE OR AFTER THE MEETING. THESE MICROCASSETTES COME WITH THE FOLLOWING PROGRAMS;
DATABASE
SPREADSHEET
WORDPROCESSOR
BUSINESS GRAPHICS
BETTER GET THEM WHILE THEY ARE AVAILABLE.

## CLASSIFIEDS

THIS CLASSIFIED SECTION IS AVAILABLE TO ALL LIST MEMBERS FREE OF CHARGE.
THE ONLY RESTRICTION IS THAT IT IS TO BE USED ONLY FOR THE SEEKING, SELLING OR SWAPPING OF SINCLAIR, TIMEX OR MICROAGE COMPUTER EQUIPMENT, PERIPHERALS AND SOFTWARE.
LISTING, LIST, AND ITS OFFICERS DO NOT ENDORSE, WARRANTY, OR GUARANTEE ANY OF THE ITEMS LISTED IN THIS CLASSIFIED SECTION

ZX-81 AND TS2040 PRINTER FOR SALE, CONTACT MR.CHARLES DOBY AT 516-842-3037.

A TS2068 SYSTEM IS FOR SALE. INCLUDES; TS2068, TS2020 TAPE RECORDER, MAGNAVOX RGB MONITOR W/INTERFACE, EPSON LX80 PRINTER, ZEBRA DOUBLE DISC SYTEM,ASSORTED PROGRAMS, BOOKS, ETC.ETC. CONTACT MR. HAL BELLINSON 36 CLINTON AVE. TROY, N.Y. 12180 TEL. 518-272-2975

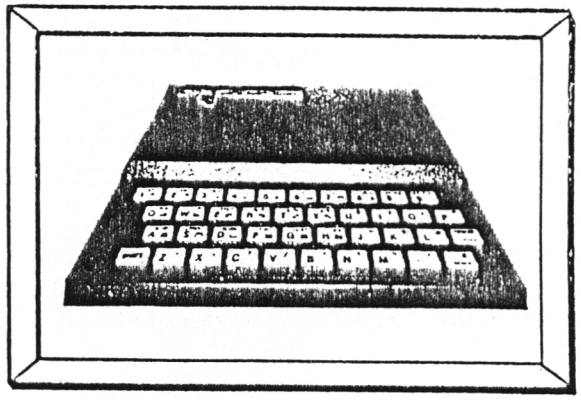
## A FINAL WORD

MY NAME IS FRED STERN, AND I AM THE EDITOR OF THIS EDITON OF LISTING.
I WOULD LIKE TO THANK JOHN PAZMINO FOR CONTRIBUTING TO THIS EDITON.
I ALSO THANK TOM SKAPINSKI FOR HIS CONTRIBUTIONS.
TS1000 TECHNICAL TIDBITS IS READY FOR SALE. SEND A CHECK FOR $4.00 PAYABLE TO LIST TO RECEIVE YOUR COPY.

LONG ISLAND GROUP
Sinclair
TIMEX

TIMEX sinclair 1000

# Long Island Sinclair-Timex Group



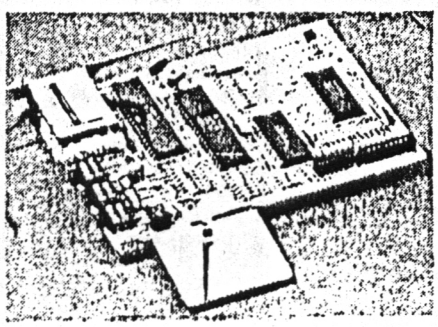
## ZX81 & T/S 1000

# TECHNICAL

# TIDBITS

T&1000 TECHNICAL TIDBIT& IS READY FOR SALE. SEND A $4.00 CHECK PAYABLE TO LIST AND A NOTE WITH YOUR NAME AND ADDRESS TO:

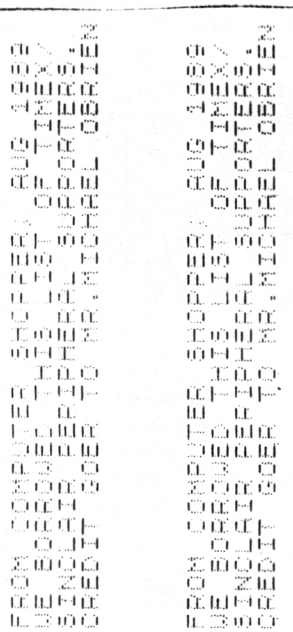
LIST
5 PERI LANE
VALLEY STREAM, N.Y. 11581

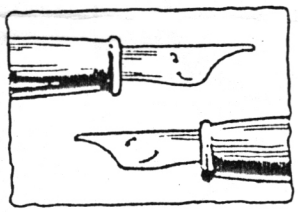


**Price $4.00**

## ◆ MORE TIMEX SINCLAIR

by Michael O'Brien





**Aerco**
Box 18093
Austin, TX 78760
512-451-5874
Custom hardware design, 1000 hardware, 2068 software. Repair

**A.F.R. Software**
1605 Pennsylvania Avenue #204
Miami Beach, FL 33139
305-531-6464
TS/2068 hardware and software

**American Micro Connection**
1256 Henderson Ave.
Sunnyvale, CA 94086
QL hardware, software and parts

**Banta Software**
8088 Highwood Way
Orangeville, CA 95662
916-722-4895
TS/1000, 2068 software

**Beaver Computer Products**
756 Fleming Avenue
Winnipeg, Manitoba
Canada R2K 1V5
204-663-2931 (evenings)
TS/2068 software

**William Bell**
596 Cherrington Road
Westerville, OH 43081
614-882-3883 (evenings)
TS/1000 software

**Paul Bingham**
PO Box 2034
Mesa, AZ 85214
TS/1000, 2068 software

**Herb Bowers Sr.**
2588 Woodshire Circle
Chesapeake, VA 23323
804-487-5924
TS/2068 software

**Brooklyn Closeout Corp.**
167 Clymer Street
Brooklyn, NY 11211
718-963-2377
TS/1000, 2068 parts, hardware and software

**Budge Robotics**
Box 18616
Tucson, AZ 85731
TS/1000, 2068 parts, repair and hardware

**Byte Back**
36 Long Terrace, PO Box 112
Leesville, SC 29070
803-532-5812
TS/1000, 2068 software and hardware

**Byte Power**
1748 Meadowview Ave
Pickering, Ontario
Canada L1V 3G8
TS/1000, 2068 software

**Cairosoft**
PO Box 10024
Hickory, NC 28602
QL, TS/2068 software

**Kurt Casby**
25 Battle Creek Court
St Paul, MN 55119
612-735-3637 (evenings)
TS/2068 software

**Church School Software**
601 North Highway 83
Bensenville, IL 60106
312-766-3848
TS/1000 software

**Curry Computer**
PO Box 5607
Glendale, AZ 85312

**Cyahoga Valley Software Works**
615 School Avenue
Cuyahoga Falls, OH 44221
216-928-7910
TS/2068 software

**C.W. Associates**
419 N. Johnson Street
Ada, OH 45810
419-634-4874
QL, Z88 hardware and software

**Jack Dohany**
390 Rutherford
Redwood City, CA 94601
415-367-7781
TS/2068 software

**Emsoft**
PO Box 8763
Boston, MA 02114
QL software

**Ezera Group II**
PO Box 5222
San Diego, CA 92105
TS/1000 software (program listings)

**Bill Ferrebee**
Mountaineer Software
749 Hill Street, #6
Parkersburg, WV 26104
304-424-7272 (evenings)
TS/2068 software

**Mike Fink**
355 W. 39th Street
New York, NY 10018
TS/2068 software

**Foote Software**
PO Box 14655
Gainsville, FL 32604
904-338-1273
TS/1000, 2068 software/hardware and publications

**Ed Grey Enterprises**
PO Box 2186
Inglewood, CA 90305
213-759-7406
TS/1000, 2068, software, hardware, parts, periph. Z88 hardware and software

**Hal-Tronix Inc.**
12671 Dix Toledo Highway
Southgate, MI 48195
313-281-7773
TS/1000, hardware, software, parts; ham radio

**Hardback Software**
Box 135
Red Rock, TX 78662
512-839-4516
TS/2068 software

**Paul Hunter**
1630 Forest Hills Drive
Okemos, MI 48864
517-349-5650
Bank switching software TS/1000

**Info-Mation**
RR#1 Box 260
California, MO 65018
314-796-4921 (evenings)
TS/1000, QL, software and hardware

**Eric Johnson**
249 N. Harden Avenue
Orange City, FL 32763
904-775-4935
TS/2068, 1000 hardware, parts and repair

**Kamrec Systems**
31274 Springlake #5304
Walled Lake, MI 48088
QL, TS/2068 software

**Ray Kingsley**
1710 Oliver Avenue
San Diego, CA 92019
TS/1000, 2068 software

**K.I.S.S. Computers**
324 Trotter Court
Sanford, FL 32773
TS/2068 software and hardware

**Knighted Computers**
12 Canalview Mall
Fulton, NY 13069
315-593-8219
TS/1000, 2068 software

**Larken Electronics**
RR #2
Navan, Ontario
Canada K4B 1H9
613-835-2680
TS/1000 software, TS/2068 software, hardware and mass storage

**Frank Lockhart**
PO Box 1131
Shelby, NC 28150
704-484-8539
TS/1000, 2068, QL software

**LST Software**
PO Box 62
Alcester, SD 57001
TS/1000 software

**Marc Fendric**
4712 Avenue "N", suite 383
Brooklyn, NY 11234
718-627-1293
QL, TS/2068, 1000 software

**McBrine Computer Products**
514 S. Jackson Street
Salisbury, NC 28144
704-633-7887 (nights)
TS/2068 hardware/software, TS/1000 software

**John McMichael**
1710 Palmer Drive
Laramie, WY 82070
307-742-4530 (nights)
TS/2068 hardware and software

**M.D.M. Enterprises**
706 South Mason
Saginaw, MI 48602
517-791-1088 (nights)
TS/1000, 2068 hardware and software

**Meta-Media Productions**
726 West 17th
Vancouver, B.C.
Canada V5Z 1T9
QL software

**Novelsoft**
35 Candel Liteway
Willowdale, Ontario
Canada M2R 3J5
416-665-0290 (nights)
TS/2068 software

**John Oliger Company**
11601 Whidbey Drive
Cumberland, IN 46229
TS/1000, 2068 hardware

**Budgetsoft**
1107 Morgan Street
Rushville, IN 46173
317-932-2431
TS/2068 software

**Promiseland Electronics**
Route 1, Box 117
Cabool, MO 65689
314-739-1712
TS/1000, 2068, QL repair

**Prospero Software LTD**
100 Commercial Street #306
Portland, ME 04101

## ◆ MORE TIMEX SINCLAIR

207-874-0382
QL Pascal/Fortran software

**Pyramid Electronics**
2174 Gulf Gate Drive
Sarasota, FL 34231
813-922-9574
QL, TS/2068, 1000 hardware and software

**Quantum Levels**
9016 Flicker Place
Columbia, MD 21045
QL, TS/1000, 2068, Z88 hardware

**Ray Rash**
2424 SW 78th Street
Oklahome City, OK 73159
TS/2068 software

**Riverside Software**
Route 1, Box B-20
Orrington, ME 04474
TS/2068 software

**RMG Enterprises**
1419 1/2 7th Street
Oregon City, OR 97045
503-655-7484
QL, Z88, TS/2068, 1000 software, hardware, parts and periph.

**Russell Electronics**
RD. 1 Box 539
Centre Hall, PA 16828
814-364-1325
QL, TS/1000, 2068 hardware and software

**Sharpe's Inc..**
Box 326
Mechanicsville, VA 23111
804-730-9697
TS/2068, Z88, QL hardware, software and repair

**Silicon Mountain Electronics**
C-12, Mountain Station Group Box
Nelson, BC
Canada V1L 2J3
TS/1000 software

**Siriusware**
6 Turning Mill Road
Lexington, MA 02173
TS/1000 software

**SSI**
424 Cumberland Avenue
Portland, ME 04101
207-761-3700
U.S. distributor of the Z88

**Sunset Electronics**
2254 Taraval Street
San Francisco, CA 94116
415-665-6161
TS/1000, 2068 software, hardware, parts and periph.

**Syncware**
602 South Mill Street
Louisville, OH 44641
216-875-1257
publication

**T&C Services**
20 Liberty Terrace
Buffalo, NY 14215
716-834-1716
QL, TS/2068, 1000 software and hardware

**Time Designs**
29722 Hult Road
Colton, OR 97017
503-824-2658
publication

**SNUG Sinclair North American Users Group**
7515 Arbordale Drive
Port Richey, FL 34668
813-863-5552


## GETTING THE IN ON THE 2068

by Dennis Jurries

The are a possible of 65536 I/O ports on the 2068. These are used by the computer for communicating with things like the keyboard or the printer, and the can be controlled from BASIC by using the IN statement.

IN is a function like PEEK
IN address

The keyboard is divided up into 8 half rows of 5 keys each.

```
IN 65278 reads CAPS SHIFT to V
IN 65022 reads A to G
IN 64510 reads Q to T
IN 63486 reads 1 to 5
IN 61438 reads 0 to 6
IN 57342 reads P to 7
IN 49150 reads ENTER to H
IN 32766 reads SPACE to B
```

The value of these addresses when no key is depressed is 31. Starting from the outside of the keyboard and moving towards the center the values are 30,29,27, 23, & 15. On some models of the Spectrum; no key depressed is 255, and the other values are 254,253,251,247, & 239.

The following program illustrates this.

```
10 FOR n=0 TO 7 REM half-row #
20 LET a=254+256*(255-2^n)
30 PRINT AT 0,0;IN a:GOTO 30
```

When you finish with each half row, press BREAK and then type NEXT n.

The IN function can be used in many ways in a program. One way is to use it to change a position of something on the screen. Say the variable x has some value: LET x=x-(IN65022=30). This will decrease the value of x by 30 only if the A key is depressed.

---

**The following Spectrum, Hints and Tips come from National Software Library 42 Harefield Road, Chean, Surrey, SM2 7NE, UK and were submitted by several of their members.**

### **MEMBER'S HINTS AND TIPS.**

So many members have sent tips for the newsletter we would need 8 pages to list them all! It's been quite a job to try and check them all and pick out the tups to include. Our thanks go to ALL who have sent tips. All the contributions published will gain free hire vouchers for the respective member.

LET a=USR 3280 or RANDOMIZE USR 3280 or LET a=USR 3582
Scrolls the screen display up one line, use a FOR-NEXT loop to scroll several lines.

LET a=USR 3330 Scrolls page to top line
LET a=USR 3583 Scrolls bottom half page, one line.
LET a=USR 3652 Clears top half of screen
POKE 23562,1 Fast auto repeat.
POKE 23561,0 Disables auto repeat
INK 9 Sets INK to contrast PAPER colour
POKE 23736,181 When making multiple SAVEs, eg with several blocks of data in a programme, add this instruction between SAVE commands. The SAVEs will then be carried out automatically without the need to 'Press a key' between each one. It sometimes helps to also use a PAUSE command, as the programmes will be very close together on the tape.
RANDOMIZE n: Is equivalent to POKE 23670,n-256xINT (n/256),POKE 23671 INT (n/256) A useful short cut to those 2 byte calculations. PEEKing those addresses will provide the neede figures without any human intervention.

An interesting and rather 'pretty' bug exists in the way the DRAW command executes it's arc making facility. Try this short routine.

```
10 INPUT "Any odd number";n
15 CLS
20 PLOT 75,50: DRAW 100,100,PI*n
30 GO TO 10
```

Having RUN this programme, try adding the following line. 5 OVER 1
Try INPUTing these numbers. 253,237,375,275,287,157, and EXPERIMENT!

POKEing the MODE system variable immediately before an INPUT command will change the character of the cursor. Try these.

POKE 23617,142 Poond sign £ (CHR$ 96)
POKE 23617,240 Dollar sign (CHR$ 36)
POKE 23617,252 Less than sign (CHR$ 60)
POKE 23617,253 More than sign (CHR$ 62) followed by less than sign (CHR$ 60)
POKE 23617,238 Flashing blank square
POKE 23617,223 Question mark
POKE 23617,190 The word USR (CHR$ 192)
POKE 23617,192 The word AIN (CHR$ 196)
POKE 23617,208 The word DATA (CHR$ 228)

By interspersing a CHR$ number between two strings for PRINTing intersting and useful relationships may be produced. i.e. PRINT "x";CHR$ z;"Y"
Set variable z as required, and change in the prog.
i.e. z = 6 is equivalent to PRINT "x","y"
z = 8 is equivalent to PRINT "x" then PRINTing "y" on top of it.
z = 9 is equivalent to PRINT "x";"y"
z = 13 is equivalent to PRINT "x""y"

```
10 PRINT (((PEEK 23732+256*PEEK 23733)+1)/1024)-16
```

This PRINTs the memory size of your SPECTRUM exactly.

POKE 23624,120 Gives a BRIGHT window at bottom of screen (Lines 22 & 23)
PRINTing to the bottom two lines may be achieved with PRINT#1; AT 0,0; (text or variable) for LINE 22
PRINT#1; AT 1,0; (text or variable) for LINE 23
(This can be used up to PRINT#1; AT 22,31; )
PRINT#2; gives normal screen PRINTing, and PRINT#3; is the same as LPRINT. Therefore, the instruction PRINT n; can be included in a programme to give bottom line printing, normal screen printing or LPRINTing, depending on the value of 'n'. This saves time and memory.
LET t=7997-USR 7997 this gives the number of 50ths of a second it took for a key to be pressed. (Just PRINT t for the result)
For a = 0 TO 255: OUT 254,a:NEXT a This causes an interesting BORDER effect.
10 BORDER 1: BORDER 2: BORDER 3: BORDER 4: BORDER 5: BORDER 6: BORDER 7: BORDER 0: PAUSE 1: GO TO 10
This also gives an interesting BORDER effect, try altering the length of the PAUSE.
LET a = USR 1278 gives a LOADing pattern. Try these other numbers as well. 1248,1251,1276,1301,1303,1305, 1306,1308,1312,1314,1315,1490,1491,1492,1488, etc. etc.
LET p= PEEK 23613-2: POKE 23613,p This will disable the BREAK key.
POKE 23692,n (n' = maximum of 255) This overides the 'Scroll' message for 'n' times.
A tip to spped up the play in 'Murder at the Manor'
1. BREAK into the programme
2. LIST 330, and BREAK scroll
3. EDIT, and add at the beginning of LINE 330 RETURN
4. Type GO TO 80 and the game will be faster, by removing the colour paint routine in the graphics.

ADDRESS by John Bzmino

This program is a most simple, quick and versatile way to manage your address book. It was inspired by an out-of-print offering from Zebra Systems, "Address Book", but it is a many-order improvement over that product.

ADDRESS has several notable features:

a) Fast positive sort routines that consume no extra memory. All sorting is done in situ

β) Screens for each function of ADDRESS are color coded to remind you where you are at all times

γ) All instructions are presented onscreen all the time to walk you thru each operation

δ) Safe recovery from accidental bombouts with all data preserved

ε) All fields in the address file are strings so you can fill them with any data you need

ζ) Address files are stored independently of the code for ADDRESS. A one loading of ADDRESS can work on any one of many different files

η) Address files are created of just the size you need for the number of records going into them. There's no large blocks of empty wasted records

θ) Operations are generally goofproof

Key in ADDRESS. You will have to alter the LOAD, SAVE, and CAT routines to suit your peculiar storage apparatus. The code as presented here is for the Portuguese disc system. The code to be examined is in the 9000 and 9500 suite of lines and in line 9999. The verbiage in lines 40 & 50 also relate to the Portuguese system and you may want to fix these up, too.

After making a safety save of ADDRESS do CLEAR: GOTO 70. You get a blue menu of all the operations in ADDRESS. Every time you complete an operation you will come back to this menu; when you load ADDRESS for action you will come to this menu first.

Now pick '11) Stop using this program'. You get a beeped warning about never ~ but NEVER ~ do RUN to recover from a bombout. (You recover by doing GOTO MENU). Right now you do GOTO 9999 to save a live copy of ADDRESS. It is this live copy that from now on you use for your address files.

In actual use the live copy of ADDRESS is loaded and you get the bluemenu. Note that at the top there is the file name and last update, but being that this here use is the first, there is no file yet loaded and so the file name is "NO FILE LOADED". Once you

create a file or you load one in off of your storage apparatus the proper file name and latest update will be displayed.

Operations from the blue menu are selected by pressing the corresponding digits for each. After each operation you may continue with it for an other round or return back to the blue menu; you are asked for your choice and you reply with "y" or "n". Many operations are logically repeatable, like searching for names. You may need, say, seven particular names in your file. So after each is turned up you have to option to "continue" — look for an other name — or "return" — stop this search function and go back to the blue menu.

If you by accident drop out of ADDRESS back into BASIC, remember, to recover your ADDRESS session you do GOTO MENU. If you do RUN you exfenestrate your address file and you must load in a spare copy — of course you did keep a spare copy, yes?

The operations are completely handheld all the way with on-screen instructions and warnings. "y/n" answers are inkeyed and strings are input. With string input you edit the entry with the normal Sinclair edit features before sending the string into ADDRESS.

Elaboration of the operations:

'0) Load address file from disc', white screen. A directory of available files on the disc is displayed and you enter the name for the desired one. Any file already resident in ADDRESS is obliterated by the loaded file. For cassette operations you will have the CAT* routines excised and the tape spotted at the proper start-mark for loading.

'1) Enter new addresses', yellow screen. The record number and count of remaining empty records are displayed. If you are running out of room, you are flashed & beeped a warning. Each field of the record is presented in turn for you to fill out.

'2) Display address file' and '3) Print address file', cyan screen, the address file is dumped to the screen or to the printer. Make sure the printer is ready. On the printer output the filename and last update are included. To interrupt a screen dump break out at the 'scroll?' query and then do GOTO MENU.

'4) Sort by last name', '5) Sort by zipcode', and '6) Sort by telephone number', blue screen. The file is physicly shuffled into the correct order and it remains in the order until you do an other sort. The progress of the sort is shown as a countdown on the screen. The sorting algorithm is self contained and no auxiliary vectors or pointers or arrays are employed.

'7) Change an address' and '8) Look for an address', magenta screen. Enter the name field to look for. Only the first few characters are really needed because the match is made on just the exact characters you type in. For a simple lookup you can get a printout.

For a change of address you are stepped thru each field. You must type in the entire new field and not just the specific alterations.

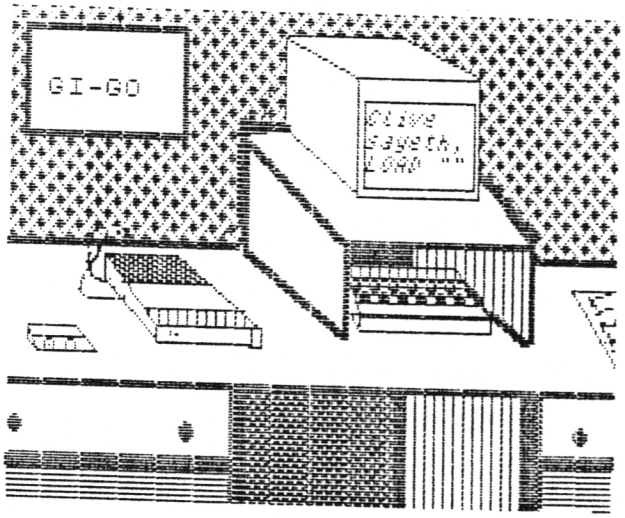
'9) Delete an address', red screen. Enter the name field like for a lookup or change. You get a confirming query before the actual deletion. A name deleted is actually cut out of the address file and all the other names are linked up to fill in the gap.

'10) Save address file to disc', green screen. You can delete/stamp the file and you then give it a name. Be aware that some storage schemes do not allow you to save a file whose name duplicates an existing name. Others do warn you of the existing file but let you overwrite it. Yet others do not care; they'll overwrite the existing file and not even tell you. You should insert routines in this 9000 block to deal with the peculiar behavior of your storage system. The file remains in ADDRESS after the save so you can continue to work with it.

'11) Stop using this program', blue screen. This merely breaks out of ADDRESS and beeps a warning at you about doing GOTO MENU to get back into ADDRESS

'12) Make a new address file', black screen. Enter the maximum number of records and the file name. Allow reasonable space for the file to grow when giving the number of records for this number is the dimension of the array constituting the address file. The upper limit is 240 records. Any file already in ADDRESS is obliterated by the new one just created.

**Quarterly Newsletter of the SOFTWARE LIBRARY**




### THAT ZERO LINE AGAIN

On reading the last issue of Computerchat, I became interested in the different methods used to create line 0's in listings. The easiest way to do this is to write a line 1 and then POKE 23756,0. However any line 0 can be removed by POKE 23756,1 or whatever.

I put my copyright message at the other end of the program. When a line 1 is written, POKE 23755,46: POKE 23756,224 will turn it into a line 2000 (in effect a line 12000). This type of line will list after 9999 and is also edit-proof as it cannot be re-entered. For POKE 23755,n, n can take any value between 40 and 63.

It is simple to write a program and then merge a short two-line BASIC program containing the copyright messages with it. However when a program of 7K or more, containing a line greater than 9999, is merged into the machine the result is a crash as the machine tries to compose a line greater than 9999 with the contents of memory. This means the auto-run cannot be avoided. If POKE 23613, PEEK 23730-5 is used to disengage the BREAK key, then the BASIC program produced can be as hard to copy as a machine code program.

The line number POKEs depend on BASIC program area starting at 23755 which is not true on a Spectrum with microdrives.

P.S. Did you know that on the Spectrum a basic line can be entered that does not contain a key word? A basic line can be entered that contains only a cursor colour command, 1-0 in extended mode....is this another bug in the ROM?

P.P.S. Try this:

```
10 PLOT 128,0
20 DRAW 0, 175, 189xPI
```

D. Spencer
Leek, Staffs.

```
  20 BORDER 0: PAPER 0: INK 7: C
LS
  30 PRINT "The new address file
 can be di- mensioned to accommo
date the ul-timate number of nam
es to be en-tered into it. The m
aximum pos- sible number of name
s is 240. Ifyou want to leave th
e file size at 240, just hit <en
ter> when   the file size is req
uested."
  34 INPUT "number of names (<=2
40): ";q$: IF q$<>"" AND VAL q$<
=240 THEN LET am=VAL q$
  36 IF q$="" OR VAL q$>240 THEN
 LET am=240
  38 LET am=am+1: DIM a$(am,104)
: LET a$(1,1)=CHR$ 0
  40 PRINT "The new address file
 name must  start with a letter
and have no more than eight char
acters. The extension '.DAT' is
attached to this name when the f
ile is sav- ed to disc."
  50 INPUT "address file name: "
;e$: IF e$(1)<"A" OR e$(1)>"z" O
R LEN e$>8 THEN GO TO 50
  60 LET a$(1,17 TO 24)=e$: GO T
O 72
  70 LET e$="NO FILE LOADED"
  72 LET y$=""
  74 POKE 23658,0: LET r$="y": L
ET a=0: LET menu=80
  80 BORDER 1: PAPER 1: INK 7: C
LS
  90 PRINT '''"******************
**************   A D D R E S S
   F I L E   ******************
**************"
  94 PRINT "address file => ";e$
''"last update => ";y$''"*********
************************"
 100 PRINT '"0) Load address fil
e from disc"'"1) Enter new addre
sses"'"2) Display address file"'
"3) Lprint address file"'"4) Sor
t by last name"'"5) Sort by zip
code"'"6) Sort by phone number"'
"7) Change an address"'"8) Look
for an address"'"9) Delete an ad
dress"
 101 PRINT "10) Save address fil
e to disc"'"11) Stop using this
program"'"12) Make a new address
 file"
 300 INPUT q$: LET y=VAL q$: IF
y>=0 AND y<=12 THEN GO TO 350
 330 GO TO 300
 350 GO SUB (9500 AND y=0)+(1000
 AND y=1)+(2000 AND (y=2 OR y=3)
)+(3000 AND (y>=4 AND y<=6))+(40
00 AND (y=7 OR y=8))+(6000 AND y
=9)+(9000 AND y=10)+(8800 AND y=
11)+(0 AND y=12)
 900 GO TO menu
1000 BORDER 6: PAPER 6: INK 0: C
LS
1002 FOR a=2 TO am: IF a$(a,1)<>
" " THEN NEXT a
1004 IF a=am THEN BEEP 1,1: BEEP
 1,1: PRINT AT 10,7; FLASH 1; PA
PER 2; INK 9;" File is full "''
FLASH 0;"no more empty addresses
 left!": PAUSE 200: RETURN
1010 IF a>=am-10 AND a<am THEN B
EEP 1,1: PRINT AT 10,7; FLASH 1;
 PAPER 2; INK 9;" File is gettin
g full "'' FLASH 0;"only ";am-a;
" empty addresses left!": PAUSE
200
1020 CLS : PRINT PAPER 5; INK 1;
" ADDRESS #";a-1'am-2-a;" empty
addresses left"
1022 INPUT "ENTER name (<= 32 CH
R$ )"'"..............................
.....";a$(a, TO 32): PRINT AT
6,0;a$(a, TO 32)
1030 INPUT "ENTER address (<= 32
 CHR$ )"'"..............................
.....";a$(a,33 TO 64): PRIN
T a$(a,33 TO 64)
1040 INPUT "ENTER city & state (
<= 22 CHR$ )"'"..................
.....";a$(a,65 TO 86): PRINT a$(
a,65 TO 86);" ";
1050 INPUT "ENTER zipcode (= 5 C
HR$ )"'"....."';a$(a,87 TO 91):
PRINT a$(a,87 TO 91)
1060 INPUT "ENTER areacode (= 3
CHR$ )"'"..."';a$(a,92 TO 94): P
RINT a$(a,92 TO 94);" ";
1070 INPUT "ENTER phone number (
<= 10 CHR$ )"'"..........";a$(a
,95 TO ): PRINT a$(a,95 TO )
1160 GO SUB 9990: IF q$="n" THEN
 RETURN
1170 NEXT a
2000 BORDER 5: PAPER 5: INK 0: C
LS
2005 IF y=2 THEN GO TO 2210
2200 LPRINT ;"==================
==============";e$;" for ";y$'"=
================================="
2205 PRINT AT 10,3; FLASH 1;" pr
inting - please wait "
2210 FOR z=2 TO a: PRINT #y;a$(z
, TO 86);" ";a$(z,87 TO 91)'a$(z
,92 TO 94);" ";a$(z,95 TO )
2220 PRINT #y;"----------------
---------------"
2230 NEXT z: GO SUB 9990: IF q$=
"n" THEN RETURN
2240 GO TO 2000
3000 CLS : PRINT AT 10,3; FLASH
1;" sorting - please wait "
3002 PRINT AT 15,0;"Sorting is f
inished when this   number steps
 down to zero =>"
3006 LET p=a: LET b=(1 AND y=4)+
(87 AND y=5)+(92 AND y=6): LET e
=(32 AND y=4)+(91 AND y=5)+(104
AND y=6)
3010 IF p<=1 THEN RETURN
3012 LET p=INT (p/2): LET m=a-p
3014 PRINT AT 16,29;p;" "
3020 LET f=0: FOR j=1 TO m: LET
k=j+p
3030 IF a$(j,b TO e)>a$(k,b TO e
) THEN LET q$=a$(j): LET a$(j)=a
$(k): LET a$(k)=q$: LET f=1
3040 NEXT j: IF f THEN GO TO 302
0
3050 GO TO 3010
4000 BORDER 3: PAPER 3: INK 7: C
LS
4010 INPUT "ENTER desired name:
"'r$
4020 PRINT AT 10,3; FLASH 1;" lo
oking - please wait "
4030 FOR z=2 TO a: IF a$(z, TO L
EN r$)=r$ THEN GO TO 4100
4040 NEXT z
4050 CLS : PRINT AT 10,(31-LEN r
$)/2;r$'TAB 11;"not found": GO S
UB 9990: IF q$="n" THEN RETURN
4060 GO TO 4000
4100 CLS : PRINT AT 6,0;a$(z, TO
 86);" ";a$(z,87 TO 91)'a$(z,92
TO 94);" ";a$(z,95 TO )
4102 INPUT "": PRINT #0;"Is this
 the right name? (y/n)": GO SUB
9991: IF q$="n" THEN GO TO 4040
4105 IF y=7 THEN GO TO 5000
4110 INPUT "": PRINT #0;"Do you
want a printout? (y/n)"
4120 GO SUB 9991: IF q$="y" THEN
 LPRINT a$(z, TO 86);" ";a$(87 T
O 91)'a$(z,92 TO 94);" ";a$(z,95
 TO )
```

```
4130 LPRINT "------------------
-------------"
4140 GO SUB 9990: IF q$="n" THEN
 RETURN
4150 GO TO 4000
5000 PRINT #0;"Change the name?
(y/n)": PRINT AT 6,0; INVERSE 1;
a$(z, TO 32)
5020 GO SUB 9991: IF q$="n" THEN
 PRINT AT 6,0;a$(z, TO 32): GO T
O 5050
5030 INPUT #0;"ENTER new name:"'
'a$(z, TO 32): PRINT AT 6,0;a$(z
, TO 32)
5050 INPUT "": PRINT #0;"Change
the address? (y/n)": PRINT AT 7,
0; INVERSE 1;a$(z,33 TO 64)
5070 GO SUB 9991: IF q$="n" THEN
 PRINT AT 7,0;a$(z,33 TO 64): GO
 TO 5100
5080 INPUT "ENTER new address:"'
'a$(z,33 TO 64): PRINT AT 7,0;a$
(z,33 TO 64)
5100 INPUT "": PRINT #0;"Change
the city-state? (y/n)": PRINT AT
 8,0; INVERSE 1;a$(z,65 TO 86)
5120 GO SUB 9991: IF q$="n" THEN
 PRINT AT 8,0;a$(z,65 TO 86): GO
 TO 5150
5130 INPUT "ENTER new city-state
:"'a$(z,65 TO 86): PRINT AT 8,0
;a$(z,65 TO 86)
5150 INPUT "": PRINT #0;"Change
the zipcode? (y/n)": PRINT AT 8,
23; INVERSE 1;a$(z,87 TO 91)
5170 GO SUB 9991: IF q$="n" THEN
 PRINT AT 8,23;a$(z,87 TO 91): G
O TO 5200
5180 INPUT "ENTER new zip code:"
'a$(z,87 TO 91): PRINT AT 8,23;
a$(z,87 TO 91)
5200 INPUT "": PRINT #0;"Change
the areacode? (y/n)": PRINT AT 9
,0; INVERSE 1;a$(z,92 TO 94)
5220 GO SUB 9991: IF q$="n" THEN
 PRINT AT 9,0;a$(z,92 TO 94): GO
 TO 5250
5230 INPUT "ENTER new area code:
 ";a$(z,92 TO 94): PRINT AT 9,0;
a$(z,92 TO 94)
5250 INPUT "": PRINT #0;"Change
the phone number? (y/n)": PRINT
AT 9,4; INVERSE 1;a$(z,95 TO )
5270 GO SUB 9991: IF q$="n" THEN
 PRINT AT 9,4;a$(z,95 TO ): GO T
O 5290
5280 INPUT "ENTER new telephone
number:"'a$(z,95 TO ): PRINT AT
 9,4;a$(z,95 TO )
5290 GO TO 4140
6000 BORDER 2: PAPER 2: INK 7: C
LS
6010 INPUT "ENTER name to be del
eted:"'r$
6015 PRINT AT 10,4; FLASH 1;" lo
oking - please wait "
6020 FOR z=2 TO a: IF a$(z, TO L
EN r$)=r$ THEN GO TO 6050
6030 NEXT z
6040 CLS : PRINT AT 10,(31-LEN r
$)/2;r$'TAB 11;"not found": GO S
UB 9990: IF q$="n" THEN RETURN
6045 GO TO 6000
6050 CLS : PRINT AT 6,0;a$(z, TO
 86);" ";a$(z,87 TO 91)'a$(z,92
TO 94);" ";a$(z,95 TO )
6054 CLS : INPUT "": PRINT #0;"I
s this the right name? (y/n)": G
O SUB 9991: IF q$="n" THEN GO TO
 6030
6056 CLS : PRINT '"If you conti
nue,"'a$(z, TO 32)'"will be de
leted"
6060 GO SUB 9990: IF q$="n" THEN
 RETURN
6080 CLS : PRINT AT 10,3; FLASH
1;" deleting - please wait "
6090 FOR z=z TO a: LET a$(z)=a$(
z+1): NEXT z: LET a=a-1
6092 GO SUB 9990: IF q$="n" THEN
 RETURN
6094 GO TO 6000
8800 BORDER 1: PAPER 1: INK 8: C
LS
8802 BEEP .5,1: BEEP .5,1: BEEP
.5,1
8804 PRINT AT 10,10;"* STOP *";A
T 15,0;"To resume, do < GO TO me
nu >"'"NEVER DO < RUN >! The f
ile willbe irretrievably destroy
ed!!"
8810 STOP
9000 BORDER 4: PAPER 4: INK 0: C
LS
9002 PRINT "You can datestamp th
e address    file by putting toda
y's date in-to it.  The date is
a string so the format  is not c
ritical.    The string is limite
d to 12 cha-racters."
9004 PRINT '"To skip the datesta
mping of the file, just hit <ent
er> when the date is requested."
9006 INPUT "ENTER today's date (
<=12 CHR$ ): "'q$: IF q$="" THEN
 LET q$="NO DATESTAMP"
9007 LET y$=q$
9008 CLS : PRINT "The address fi
le now has the    name:";TAB 10;
e$'"If you change it, the new na
me  must begin with a letter and
    have no more than 8 characte
rs. To leave the name alone just
 hit<enter> when the name is req
ues-ted."
9012 INPUT "ENTER file name (omi
t '.DAT'):"'q$: IF q$="" THEN GO
 TO 9018
9015 IF q$(1)<"A" OR q$(1)>"z" O
R LEN q$>8 THEN GO TO 9012
9016 LET e$=q$
9018 PRINT '"  SET UP THE DISC
FOR WRITING"
9020 GO SUB 9990: IF r$="n" THEN
 RETURN
9040 CLS : PRINT AT 10,8; FLASH
1;" SAVING ";e$;".DAT"
9045 LET a$(1,14 TO 16)=STR$ am:
 LET a$(1,17 TO 24)=e$: LET a$(1
,2 TO 13)=y$: LET a$(1,25 TO 27)
=STR$ a
9050 SAVE *e$+".DAT" DATA a$():
RETURN
9500 BORDER 7: PAPER 7: INK 0: C
LS
9510 PRINT "**********************
************  List of address fi
les (+.DAT) **********************
************"
9520 CAT *: INPUT "file name (om
it '.DAT')"'e$
9522 CLS : PRINT AT 10,8; FLASH
1;" LOADING ";e$;".DAT"
9530 LOAD *e$+".DAT" DATA a$()
9532 LET a=VAL a$(1,25 TO 27): L
ET am=VAL a$(1,14 TO 16): LET e$
=a$(1,17 TO 24): LET y$=a$(1,2 T
O 13)
9534 RETURN
9990 PRINT #0;"PRESS <y> to cont
inue"'"PRESS <n> to return to me
nu"
9991 PAUSE 0: LET q$=INKEY$: IF
q$<>"y" AND q$<>"n" THEN GO TO 9
991
9992 INPUT "": RETURN
9999 SAVE *"address" LINE 70
```

# ADD EEPROM TO YOUR COMPUTER

*Electrically erasable PROMs are fast, random access, don't forget when power is lost, & easily reprogrammed*

*By Michael Keryan*

SEVERAL memory devices involving new technologies have become available in the last few years. Among them are the electrically alterable read-only memory (EAROM), the electrically erasable programmable read-only memory (EEPROM), and the nonvolatile random access memory (NOVRAM), all of which show great promise for replacing conventional memory chips in certain applications. They have the attributes of the ideal memory: they are fast, random access; they don't "forget" everything when power is lost; and they can be reprogrammed without removing them from their sockets in the computer.

Since they are several times more expensive than conventional memory chips, the use of these new memory devices is not widespread in current circuit designs. Owing to their advantages, however, and the likelihood that prices will be reduced (as is historical in this industry), it's appropriate to get a leg up on this technology. Therefore, let's examine such a chip, using as an example the X2816A by Xicor, Inc., which is extremely easy to interface. The X2816 is a 2K X 8 (2048-byte) EEPROM that runs on a single 5-volt power supply. The chip requires no external latches or data preconditioning and can be used like RAM. Since it has the same basic pinout as the popular 2716 EPROM, 2316 ROM, and 6116 RAM, the X2816 can be placed in sockets for any of these chips, with only minor rewiring required for the control signals. The chips are ideal for the new 2716 EPROM/6116 RAM boards now available for a number of computers.

**About the Chip.** The pinout of the X2816 is shown in Fig. 1. Three input pins control the action of the read and write cycles: chip enable ($\overline{CE}$), output enable ($\overline{OE}$), and write enable ($\overline{WE}$). A low signal to $\overline{CE}$ switches the chip from the low-current (less than 40 mA) standby mode to the active (100 mA) mode. This $\overline{CE}$ line must be brought low to either read or write to the chip.

To read data from the chip, the $\overline{OE}$ line is brought low (while $\overline{WE}$ is held high). To write data to the chip, the $\overline{WE}$ line is brought low (while $\overline{OE}$ is held high). The data input/output lines are inhibited in the other possible states of the control lines. To summarize, $\overline{OE}$ can be considered as a read pin and $\overline{WE}$ a write pin. A low pulse to one of them results in a read or write cycle, provided the $\overline{CE}$ is also low. These operation modes are shown in the Table.

The X2816 has on-chip latches, a timing-pulse generator, and a high-voltage pulse generator used for the write cycle. The write cycle takes up to 10 milliseconds to complete, but once triggered (by $\overline{CE}$ and $\overline{WE}$ low and $\overline{OE}$ high), the cycle is automatic. Therefore, the X2816 can be used as RAM, but the chip should not be accessed until at least 10 milliseconds after a write cycle has been initiated. When not writing to the chip, the read access time is similar to conventional memory chips (300 nano-seconds or less). The chips contain write-protection circuitry for power-up/power-down transitions and glitches on the WE pin. They are designed for 10,000 write cycles.

**OPERATION MODES OF THE X2816**

| Control Lines | | | Mode of operation |
|---|---|---|---|
| $\overline{CE}$ | $\overline{OE}$ | $\overline{WE}$ | |
| H | L | L | Low-power standby |
| H | L | H | Low-power standby |
| H | H | L | Low-power standby |
| H | H | H | Low-power standby |
| L | L | L | Inhibit |
| L | L | H | Read cycle, data = output |
| L | H | L | Write cycle, data = input |
| L | H | H | Inhibit |

**Wiring the EEPROM.** In many cases, normal 6116 CMOS static RAM chips can be replaced by X2816 EEPROMs with no changes required to the wiring. The pin-outs of the X2816 and 6116 are identical. As shown in Fig. 2A, the memory read line is brought to $\overline{OE}$, the memory write line is brought to $\overline{WE}$, and the chip enable line is brought to $\overline{CE}$. For new circuitry using the Z80, 8085, 8080, etc., the interfacing is straightforward. CPU chips such as the 6502, 6809, 6800, etc., usually use a single read/write line (R/$\overline{W}$) that is high on a read cycle and low on a write cycle. In this case, the $\overline{OE}$ input can be derived by inverting the R/$\overline{W}$ line connected to the $\overline{WE}$ input, as shown in Fig. 2B. In any case, when using the EEPROM as RAM, you will probably want to address it outside of the normally used contiguous block of workspace RAM, to protect it from being written to indiscriminately. This can be accomplished either by hardware (decoders address it away from the other RAM) or software techniques (setting the boundaries of the workspace RAM to exclude the EEPROM).

A simple method of adding EEPROM to a TRS-80 is shown in Fig. 3. This basic addressing scheme is used to add either RAM or EPROM addresses in address space $3000 to $37DF. Note that the last 32 bytes of the EEPROM will not be used. The Disable switch allows the EEPROM to be write-protected, and the LED lights when the EEPROM can be written to.

When used to replace EPROM or ROM in an existing circuit, some part has to be changed to allow the chip to be active on both the read and write cycles. Normally, EPROM and ROM chips are enabled only on read cycles. After defeating the read-only access, wire the $\overline{OE}$ and $\overline{WE}$ as previously explained.

To "burn" the chip, merely write data to a valid address by using the BASIC POKE statement, or use a machine language or assembly program to move or store data in the desired address. If more than one byte will be written, the program must provide for the approximate 10 milliseconds dead time after each write cycle. Waste-time loops can easily be added to both BASIC and assembler programs. An alternate method is to verify each byte prior to writing the next byte; the data lines will be in an open state during the internally timed write cycle. However, since reading data from an open-state data bus will result in a certain data byte (dependent on type of computer, usually $00 or $FF), when writing this certain data byte be sure to insert a software timing loop.

**Applications.** The EEPROM can be used in just about any application where conventional EPROM or ROM is now used. The difference is that now you will be able to easily change the code. Use an EEPROM for a character generator, bootstrap programs, input/output drivers, new utility programs, to fix bugs in your BASIC-in-ROM, etc. It can also be used in applications that were impossible to do before. Use it to download program or data files from EEPROM to RAM and then save the updated data back to EEPROM; that is, a disk emulator with non-volatile memory. Remember, the 10,000 write cycle lifetime will allow daily updates for over 25 years.

For more information on EEPROMs, write to Xicor, Inc., 851 Buckeye Court, Milpitas, CA 95035. Ask for application note #104 and a data sheet for the X2816A, which is currently priced at about $35 for single units. Other EEPROM manufacturers include General Instruments, Hitachi, Intel, National, Motorola, and Seeq. Seeq also has a 5 volt-only EEPROM available.◇

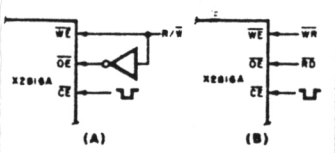



*Fig. 2. Interfacing to the X2816 depends on the type of CPU used.*

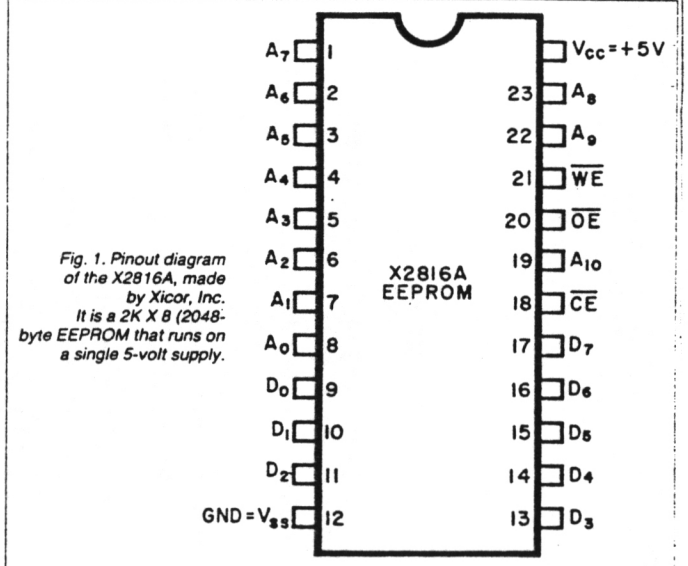



*Fig. 1. Pinout diagram of the X2816A, made by Xicor, Inc. It is a 2K X 8 (2048-byte EEPROM that runs on a single 5-volt supply.*

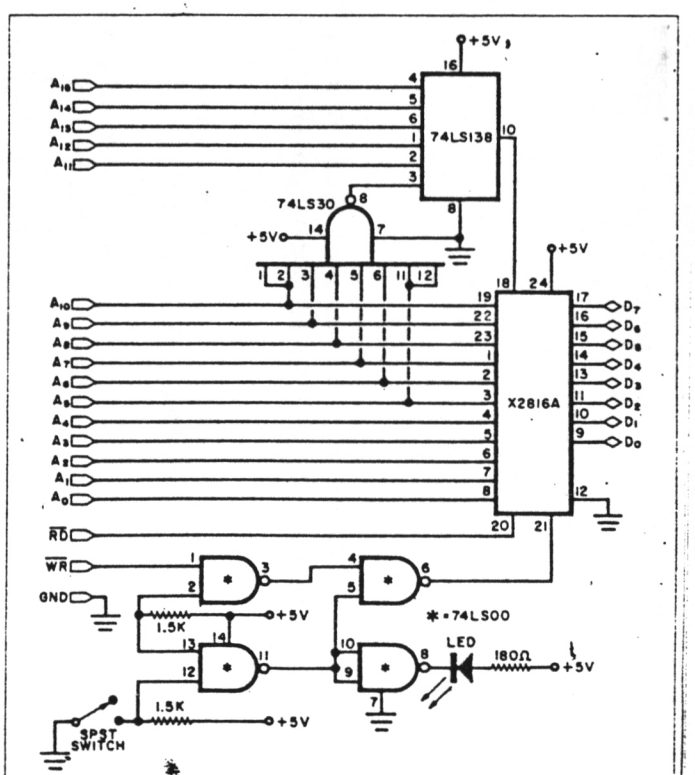



*Fig. 3. A simple method of adding EEPROM to a TRS-80.*

## LISTinG Policy

Annual Dues.....$15.00

One "Sample" copy sent upon receipt of large SASE.

Copies Provided on exchange basis with other bona fide user groups.

**L.I.S.T.inG** is published monthly by **LIST** (Long Island Sinlair Timex) Group, a non-for-profit users group.

### NOTE:

Normal membership year is Feb. through Jan. at a cost of U$15.00. By keeping as many members as possible on that basis, we keep our costs and chances of error down. If you wish to begin your subscription later in the year, please sign up for the end of this year and all of next.

We will accept partial years or different sucription runs, on a limited basis ( particularly from members outside the U.S.). But please be aware that in addition to possible rate increases, your "account" must be handled "by hand" and errors may occur.

### POLICY REGARDING CONTRIBUTED MATERIAL:

We are always looking for interesting articles, programs, reviews etc. to keep our members informed and entertained. Articles submitted for publication are printed on the following basis :

1) You , the writer, maintain the full copyright and can resell, lend or give away your work, as you wish.

2) We are granted the right to publish your material, in the original issue in which it appears. Reprints (e.g., to supply orders for back issues) will include your material as a part of its original issue.

We can't (for now) pay you for your material, but you will receive a copy of the issue in which it is published, even if you are not a member. You may get more than one issue and you will definitely earn the respect and appreciation by your grateful peers.



Articles represent the opinion of the author and not necessarily the **LIST** Group. **LIST** disclaims any resposibility for any thing you may do to your computer as a result of reading any articles in **LISTinG**.

Please note our new address : LIST
5 PERI LANE
VALLEY STREAM, N.Y. 11581